

**REMARKS**

Applicant respectfully requests reconsideration and allowance of the subject application. Claims 13-24 are pending, of which claims 22-24 have been amended. The amendment to claims 22-24 are simply to correct an informality, and not to overcome prior art.

**35 U.S.C. §112 Claim Rejections**

Claims 22-24 are rejected under 35 U.S.C. §112, second paragraph, as being indefinite (*Office Action* p.3). Appropriate amendments to claims 22-24 have been provided herein and Applicant respectfully requests that the §112 rejection be withdrawn.

**35 U.S.C. §102 Claim Rejections**

Claims 13-24 are rejected under 35 U.S.C. §102(b) as being anticipated by Hendren et al., "Supporting Array Dependence Testing for an Optimizing/Parallelizing C Compiler" (1993) (hereinafter, "Hendren") (*Office Action* p.4). Applicant respectfully traverses the rejection.

As the title indicates, Hendren describes features for a C compiler, such as support analyses in the context of C parallelizing compilers to handle loops and aliasing due to pointers (*Hendren* p.2, ¶1). Specifically, Hendren describes a framework designed to handle the complexities of the C language that affect array dependence analysis (*Hendren* p.3, ¶1.2). Hendren illustrates in Fig. 2 and describes that C program files are input to a compiler which produces a simplified

1 and compositional structured representation called "SIMPLE" to handle various  
2 complications in the *C* program files (*Hendren* p.4, ¶2; Fig. 2).

3 In the Background section of the present application, Applicant describes  
4 compiling a high-level programming language into machine instructions where the  
5 compiling includes lexical, syntax, and semantic analysis, as well as generating a  
6 syntax tree during compiling (*Specification* p.1, line 19 to p.2, line 18). In contrast  
7 to a compiler that translates a program written in a specific programming language  
8 (e.g., the *C* programming language) into another programming language or  
9 machine code, as described in *Hendren*, Applicant describes that a program can be  
10 developed and represented by a "high-level program tree that is a  
11 syntax-independent representation" of a programmer's intent (*Specification* p.4,  
12 lines 33-37). A programmer can directly manipulate the syntax-independent  
13 program tree, "which is in contrast to conventional programming systems in which  
14 a programmer manipulates a textual representation of the program that is later  
15 converted into a syntax tree during compilation" – as described in *Hendren*.  
16 (*Specification* p.5, lines 1-5).

17  
18 Claim 13 recites a method comprising "identifying a syntax-independent  
19 programming intent represented as a first node of a data structure". *Hendren* does  
20 not show or disclose a syntax-independent programming intent, or identifying a  
21 syntax-independent programming intent represented as a node of a data structure,  
22 as recited in claim 13.

23 As described above, *Hendren* is syntax-dependent and describes an  
24 intermediate structure to compile *C* program files (*Hendren* p.4, ¶2; Fig. 2). The  
25

1 Office cites Hendren for teaching a syntax-independent programming intent  
2 represented as a first node of a data structure (*Office Action* p.4). Applicant  
3 disagrees because what Hendren illustrates in Fig. 7(b) is a general expression tree  
4 that represents an index in an example simplified program illustrated in Fig. 7(a)  
5 (*Hendren* pp.9-10, ¶5.1; Fig. 7). Hendren describes that a general expression tree  
6 is built during the structured "SIMPLE" analysis when compiling the C program  
7 files (*Hendren* p.4, ¶2; Fig. 2; p.9, ¶5.1).

8 As described in Hendren, the general expression tree is generated from a  
9 source program during compiling. To the contrary, Applicant claims identifying a  
10 syntax-independent programming intent represented as a data structure node.  
11 Accordingly, claim 13 along with dependent claims 14-16 are allowable over  
12 Hendren and Applicant respectfully requests that the §102 rejection be withdrawn.

13  
14 Claim 17 recites a method of handling data comprising "reading a syntax-  
15 independent programming intent represented as a first node of a hierarchical tree".  
16 Hendren does not show or disclose a syntax-independent programming intent, or  
17 reading a syntax-independent programming intent represented as a node of a  
18 hierarchical tree, as recited in claim 17.

19 As described above in the response to the rejection of claim 13, Hendren is  
20 syntax-dependent and describes an intermediate structure to compile C program  
21 files (*Hendren* p.4, ¶2; Fig. 2). Further, Hendren describes that a compiler  
22 generates a general expression tree from a source program during compiling  
23 (*Hendren* p.4, ¶2; Fig. 2; pp.9-10, ¶5.1). Hendren also does not disclose a syntax-  
24 independent programming intent, as recited in claim 17.  
25

1 Further, the Office rejects claim 17 as having corresponding functionality  
2 to claim 13 (*Office Action* p.7). Applicant disagrees that “reading a syntax-  
3 independent programming intent”, as recited in claim 17, is a corresponding  
4 function to “identifying a syntax-independent programming intent”, as recited in  
5 claim 13. The Office has not provided a basis for the rejection of “reading a  
6 syntax-independent programming intent” other than to state vaguely that this is  
7 “corresponding functionality” (*Office Action* p.5).

8 Accordingly, claim 17 along with dependent claims 18-20 are allowable  
9 over Hendren and Applicant respectfully requests that the §102 rejection be  
10 withdrawn.

11  
12 Claim 21 recites a data structure comprising “a first node received as an  
13 input and configured for display as a representation of a syntax-independent  
14 programming intent”. Hendren does not show or disclose a node of a data  
15 structure “received as an input and configured for display as a representation of a  
16 syntax-independent programming intent”, as recited in claim 21.

17 As described above in the response to the rejection of claim 13, Hendren is  
18 syntax-dependent and describes an intermediate structure to compile C program  
19 files (*Hendren* p.4, ¶2; Fig. 2). Hendren describes that a compiler generates a  
20 general expression tree from a source program during compiling (*Hendren* p.4, ¶2;  
21 Fig. 2; pp.9-10, ¶5.1).

22 Further, Hendren does not show or disclose that a node of a data structure  
23 can be received as an input, or that a node received as an input is configured for  
24 display. Although Hendren illustrates the general expression tree in Fig. 7(b),  
25

1 there is no indication whatsoever that the general expression tree is configured for  
2 display. Accordingly, claim 21 along with dependent claims 22-24 are allowable  
3 over Hendren and Applicant respectfully requests that the §102 rejection be  
4 withdrawn.

5  
6 **Conclusion**

7 Pending claims 13-24 are in condition for allowance. Applicant  
8 respectfully requests reconsideration and issuance of the subject application. If  
9 any issues remain that preclude issuance of this application, the Examiner is urged  
10 to contact the undersigned attorney before issuing a subsequent Action.

11  
12 Respectfully Submitted,

13  
14 Dated: Jun 15, 2004

15 By: 

16 David A. Morasch  
17 Reg. No. 42,905  
18 (509) 324-9256 x 210  
19  
20  
21  
22  
23  
24  
25